

LATIHAN:**1. Penjelasan Method SelectAllStudents**

- Menghubungkan method selectCourses dengan method SelectAllStudents, yaitu dengan menambahkan anotasi Results dengan variable property, column, javaType, dan many.

```
@Select("SELECT npm, name, gpa FROM student")
@Results(value = {
    @Result(property="npm", column="npm"),
    @Result(property="name", column="name"),
    @Result(property="gpa", column="gpa"),
    @Result(property="courses", column="npm",
        javaType = List.class,
        many = @Many(select="selectCourses"))
})
List<StudentModel> selectAllStudents();
```

- Anotasi Results digunakan untuk memetakan hasil dari query select (npm, name, gpa dari seluruh student) ke class StudentModel

- Variabel property ="courses" artinya kita ingin mengisi variable courses yang ada pada class StudentModel. Variabel column="npm" artinya kolom npm pada database akan dijadikan parameter untuk method selectCourses. Variabel javaType menentukan class yang menjadi kembalian. Kemudian variabel many diset dengan method selectCourses yang akan mengisi variabel courses.

- Method selectCourses sudah dibuat sebelumnya. Method ini akan mengembalikan list courses yang diambil oleh student dengan npm tertentu.

```
@Select("SELECT course.id_course, name, credits " +
    "FROM studentcourse join course " +
    "ON studentcourse.id_course = course.id_course " +
    "WHERE studentcourse.npm = #{npm}")
List<CourseModel> selectCourses(@Param("npm") String npm);
```

- Menambahkan keterangan view pada viewall.html

```

<h1>All Students</h1>

<div th:each="student, iterationStatus: ${students}">
  <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
  <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
  <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
  <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
  <h3>Kuliah yang diambil</h3>
  <ul th:each="course, iterationStatus: ${student.courses}">
    <li th:text="${course.name} + ' - ' + ${course.credits} + ' sks'">
      Nama kuliah-X SKS
    </li>
  </ul>
  <a th:href="'/student/delete/' + ${student.npm}">Delete Data</a><br/>
  <a th:href="'/student/update/' + ${student.npm}">Update Data</a><br/>
  <hr/>
</div>
</body>

```

2. Penjelasan menambahkan View pada Course

- Membuat method selectStudentCouses pada kelas StudentMapper

```

@Select("SELECT student.npm, name " +
        "FROM studentcourse join student " +
        "ON studentcourse.npm = student.npm " +
        "WHERE studentcourse.id_course = #{id_course}")
List<StudentModel> selectStudentCourses();

```

Method ini mengembalikan student yang mengambil course berdasarkan pencarian suatu course

- Membuat method selectCourse pada kelas StudentMapper

```

ntCont...  StudentServi...  *StudentMap...  StudentServi...  viewcourses...  »_2
@Select("SELECT course.id_course, name, credits " +
        "FROM studentcourse join course " +
        "ON studentcourse.id_course = course.id_course " +
        "WHERE studentcourse.npm = #{npm}")
List<CourseModel> selectCourses(@Param("npm") String npm);

@Select("SELECT id_course, name, credits FROM course WHERE id_course = #{id_course}")
@Results(value = {
    @Result(property="id_course", column="id_course"),
    @Result(property="name", column="name"),
    @Result(property="credits", column="credits"),
    @Result(property="students", column="id_course",
        javaType = List.class,
        many = @Many(select="selectStudentCourses"))
})
CourseModel selectCourse(@Param("id_course") String id_course);

```

- Menambahkan method selectStudentCourses dan selectCourse pada kelas StudentService

```
3 import java.util.List;
4
5 import com.example.tutorial5.model.CourseModel;
6 import com.example.tutorial5.model.StudentModel;
7
8 public interface StudentService {
9     StudentModel selectStudent(String npm);
10
11     List<StudentModel> selectAllStudents();
12
13     void addStudent(StudentModel student);
14
15     void deleteStudent(String npm);
16
17     void updateStudent(StudentModel student);
18
19     CourseModel selectCourse(String id_course);
20
21     List<StudentModel> selectStudentCourses();
```

- Membuat method selectStudentCourses dan method selectCourse pada kelas StudentServiceDatabase

```
StudentCont... StudentServi... StudentMapp... StudentServi...
41     }
42
43
44 public void updateStudent(StudentModel student){
45     studentMapper.updateStudent(student);
46     log.info("student " + student.getNpm() + " updated")
47 }
48
49 public List<StudentModel> selectStudentCourses(){
50     log.info("select all courses with the students");
51     return studentMapper.selectStudentCourses();
52 }
53
54 public CourseModel selectCourse(String id_course){
55     log.info("select course with id {}", id_course);
56     return studentMapper.selectCourse(id_course);
57 }
58
59
```

- Membuat method viewCourses pada kelas StudentController untuk sebagai view ketika menampilkan suatu courses beserta student

```

    }

    @RequestMapping("/course/view/{id_course}")
    public String viewCourses (Model model,
        @PathVariable(value = "id_course") String id_course)
    {
        CourseModel course = studentDAO.selectCourse(id_course);
        if (course != null) {
            model.addAttribute ("course", course);
            return "viewcourses";
        } else {
            model.addAttribute ("id_course", id_course);
            return "not-found-student";
        }
    }
}

```

- Membuat tampilan view yaitu viewcourses.html

```

StudentCont... StudentServi... StudentMapp... StudentServi... viewcourses.... »
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View All Students</title>
5 </head>
6 <body>
7 <h3 th:text="'ID = ' + ${course.id_course}">ID Course</h3>
8 <h3 th:text="'Nama = ' + ${course.name}">Name</h3>
9 <h3 th:text="'SKS = ' + ${course.credits}">Credits</h3>
10 <h3>Mahasiswa yang mengambil</h3>
11 <ul th:each="student, iterationStatus: ${course.students}">
12 <li th:text="${student.npm} + ' - ' + ${student.name}">
13 NPM student-Name student
14 </li>
15 </ul>
16 <hr/>
17 </body>
18 </html>
19

```

- Hasil program



ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang mengambil



ID = CSC124

Nama = SDA

SKS = 3

Mahasiswa yang mengambil

- 1506689263 - agnes
 - 150666060 - maria
-

LESSON LEARNED:

Saya menjadi mengetahui bagaimana cara menggunakan database serta relasi pada database dalam project Spring Boot. Saya menjadi mengerti bagaimana cara membuat relasi antar suatu entitiy di dadatabase. Saya juga menjadi mengetahui variable baru seperti variable property, column, javaType, dan many.